A TOPOLOGY DISCOVERY ALGORITHM FOR SENSOR NETWORKS WITH APPLICATIONS TO NETWORK MANAGEMENT (SHORT PAPER)

Budhaditya Deb, Sudeept Bhatnagar and Badri Nath

Rutgers University, NJ, 08854.
`{bdeb,sbhatnag,badri}@cs.rutgers.edu`

ABSTRACT

In this paper we describe a topology discovery algorithm *(TopDisc)* for wireless sensor networks with its applications to network management. The algorithm finds a set of distinguished nodes, using whose neighborhood information we can construct the approximate topology of the network. Only these distinguished nodes reply back to the topology discovery probes, thereby reducing the communication overhead of the process. These nodes logically organize the network in the form of clusters comprised of nodes in their neighborhood. *TopDisc* forms a Tree of Clusters *(TreC)* rooted at the monitoring node, which initiates the topology discovery process. We show how managing a complex network of sensor nodes is simplified using a *TreC*. This organization is used for efficient data dissemination and aggregation, duty cycle assignments and network state retrieval. The *TreC* compares well to a centralized approach and is optimal in the number of hops from the monitoring node. The mechanisms proposed are completely distributed, use only local information and use only one packet per node and thus are well suited for sensor networks.

1. INTRODUCTION

In this paper, we propose an efficient Topology Discovery Algorithm (*TopDisc)* for sensor networks with its applications to data dissemination and aggregation, duty cycle assignments and network state retrieval. Network Topology is an important model of the network state as it implicitly gives a lot of information about the active nodes present and the connectivity/reachability map of the system. Many topology discovery algorithms exist for wired networks but most of them introduce a lot of additional traffic into the network [8]. Since sensor networks are energy-constrained [1-3], the protocols used in sensor networks have to be energy-efficient[4-6].

The *TopDisc* algorithm uses the fact that wireless is a broadcast medium of communication. A node can know the existence of other nodes in its communication range just by listening to the communication channel. The algorithm takes advantage of this fact to find a set of distinguished nodes using whose neighborhood information we can construct the approximate topology of the network. Only distinguished nodes reply back to the topology discovery probes, thereby reducing the communication overhead of the process. These distinguished nodes form clusters comprised of nodes in their neighborhood. These clusters are arranged in a tree structure called *TreC,* rooted at the monitoring node (or the initiating node).

The *TreC* represents a logical organization of the nodes and provides a framework for managing sensor networks. Using only local information between adjacent clusters, information flows from nodes in one cluster to nodes in a cluster at a different level in the *TreC*. The clustering also provides a mechanism to assign node duty cycles such that a minimal set of nodes are active in maintaining the network connectivity. The cluster heads incur only minimal extra overhead of setting up the structure and maintaining local information about its neighborhood.

2. OVERVIEW OF TOPOLOGY DISCOVERY

In this section we describe the topology discovery algorithms used in sensor networks. We divide topology discovery in three stages of execution.

- A monitoring node requiring the topology of the network initiates a "topology discovery request".
- This request diverges throughout the network reaching all active nodes.
- A response action is set up which converges back to the initiating node with the topology information.

We assume that the request divergence is through controlled flooding so that each node forwards exactly one "topology discovery request". Note that each node should send out at least one packet for other nodes to know its existence. This also ensures that all nodes receive a packet if they are connected. However various methods may be employed for the response action. The different mechanisms described, differ only in this aspect and affect the overall efficiency of the process.

We use two trivial approaches to compare the performance of our approach. They are:

1. *Direct Response:* When a node receives a topology discovery request, it forwards this message and immediately sends back a response with its neighborhood list along the reverse path.
2. *Aggregated Response:* A node receives a packet, it forwards the request immediately but waits for its children nodes to respond before sending its own response. On receiving responses from its children, it aggregates the data and sends it to its own parent.

3. TOPDISC ALGORITHM

In the trivial approaches, all nodes respond back to the topology discovery queries. *TopDisc* differs from the trivial approaches in its response mechanism. Only a subset of nodes is selected to respond back to the topology discovery queries. The union of neighborhood lists of the selected subset of nodes forms the approximate topology of the network.

The subset chosen is such that each node in the network is either a part of the subset or is a neighbor of a node in the subset. Thus the subset is a dominating set for the network and should have minimum cardinality for optimal consumption of resources. We describe a coloring algorithm that finds an approximate solution to the above problem in a distributed manner and compares well to a centralized solution of the same. The algorithm is described below:

A. The Coloring Algorithm to find the Responding Set.

We use three colors to select the responding set. The different colors and their definitions are given below. All nodes, which receive a topology discovery request packet and are alive, to respond are considered as discovered nodes.

- ***White:*** Yet undiscovered node, or node, which has not received any topology discovery packet.
- ***Black*:** Cluster head node, which replies to topology discovery request with its neighborhood set.
- ***Grey*:** Node which is covered by at least one black node i.e. it is neighbor of a black node.

The algorithm is as follows:

- The node which initiates the topology discovery request is colored black and broadcasts a topology discovery request packet.
- All white nodes become grey nodes when they receive a packet from a black node. Each grey node broadcasts the request to all its neighbors with a random delay inversely proportional to its distance from the black node from which it received the packet.
- When a white node receives a packet from grey node, it becomes a black node with some random delay. In the meantime if it receives any packet from some other black node, it becomes a grey node. Again the random delay is inversely proportional to the distance from the grey node from which the request was received.
- Once nodes are grey or black, they ignore other topology discovery request packets.

Initially all nodes are white. When the topology discovery request propagates, each node is colored black or grey according to their definition. At the end of the initial phase of the algorithm, each node in the network is either a black node or a neighbor of a black node (i.e. grey node). All nodes broadcast a topology discovery request packet exactly once in the initial phase of the algorithm. Thus all nodes have the neighborhood information just by listening to these transmissions. Hence, nodes have the neighborhood lists available before the topology acknowledgement is returned. We use the forwarding delay heuristic to approximate a greedy approach to find the subset.

B. TopDisc Response Mechanism

The first phase of the algorithm sets up the node colors. The initiating node becomes the root of the black node tree where the parent black nodes are at most two hops away. Each node has the following information at the end of this period:

- A grey node knows its neighboring black node.
- Each node knows its *parent black node*, which is the last black node from which the topology discovery was forwarded to reach it.
- Each black node knows the default node to which it should forward packets to reach the parent black node. This node is essentially the node from which it received the topology discovery request.
- All nodes have their neighborhood information.

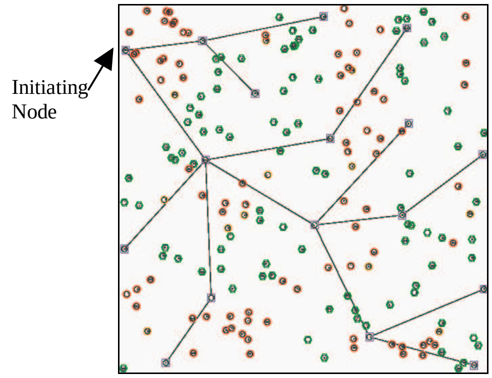

Figure 1. Illustration of typical TreC for 200 nodes and range 20m

Using the above information, the steps for *TopDisc Response* are as follows:

- When a node becomes black, it sets up a timer to reply to the discovery request. Each black node waits for this time period during which it receives responses from its children black nodes.
- It aggregates all neighborhood lists from its children and itself and when its time period for acknowledgement expires, forwards the aggregated neighborhood list to the default node to its parent.
- All forwarding nodes in between black nodes may also add their adjacency lists to the list from black nodes.

Here we note that for the algorithm to work properly, timeouts of acknowledgements should be properly set. For example timeouts of children black nodes should always expire before a parent black node. For this we set a timeout value inversely proportional to the number of hops a black node is away from the monitoring node. The aggregation tree thus obtained is termed a *TreC* (Tree of Clusters). A typical *TreC* obtained by *TopDi*sc is shown in figure 1. The example shows a 100x100 sq.m. area with 200 nodes and communication range 20m. The arrow represents the initiating node.

We note the following characteristics of the TopDisc Algorithm.

- The total surface area and the communication range of nodes bound the maximum number of black nodes formed and is almost constant with respect to density of the network.
- The timer mechanisms ensure that the TreC is optimal in the number of hops i.e. each black node is optimal number of hops away from the monitoring node in the *TreC.*

4. APPLICATIONS OF TOPOLOGY DISCOVERY

In this section, we describe some applications of *TopDisc* to certain sensor network management issues.

- *Retrieving Network State*
  - *Reachability Map:* The *TopDisc* mechanism provides a reachability map of the region. Note that a complete connectivity map (comprising all the edges in the network) is a superset of the reachability map. Direct response and aggregated response mechanism provide connectivity map.
  - *Energy Map:* If the neighborhood information consists of the energy of neighboring nodes, approximate energy map can be retrieved efficiently using TopDisc.
  - *Usage Model:* Like in the previous case, each node can transmit number of bytes received and transmitted by it during the last *t* minutes. A black node will have this information cached at the time it sends its response.
- *Data Dissemination and Aggregation:* We believe that in sensor networks all information flow would be from sensors to some monitoring node with some control information being transmitted from monitoring node to sensors. The topology discovery process sets up a *TreC* rooted at the initiating node. Since the TreC is optimal in the number of hops from the monitoring node, it can be used for efficient data dissemination and aggregation.

The above illustrates the direct use of TopDisc and TreC in various scenarios. We now propose a mechanism to set up forwarding node duty cycles using the *TreC*.

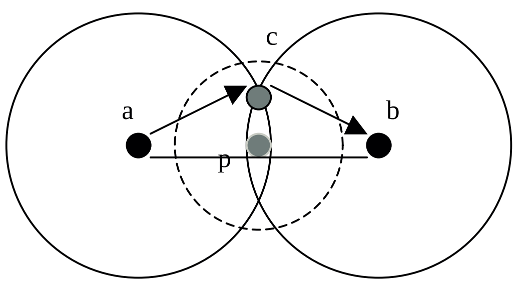

Figure 2. Assigning up the duty cycle with location information..

A. *Duty Cycle Assignment*

Each node in any cluster has at least two kinds of information: *neighboring black node* and the *parent black node*, which is the last black node from which, the topology discovery request had been forwarded. In each cluster, using this information, sets of nodes between two clusters are chosen which can forward packets between clusters. At least one node in each set is active at a given time to maintain a link between a parent/child cluster pair.

Figure 2 shows a general case in which a cluster (with black node *b*) may be formed as child of another cluster (with black node *a*). The communication range of nodes is equal to R. If we consider a circular area with radius R/2 (shown by the dotted circle), nodes inside this region would always form a completely connected graph, as each would be within communication range of others. We consider such a region centered at the midpoint (point *p*) of a parent/child cluster pair. If there is at least one node active in both clusters inside this region, then there is always a way to forward a packet from one cluster to the other. The various steps for setting up the sets of nodes are given below:

- Black nodes broadcast the location of parent and children clusters in their neighborhood.
- Nodes decide to be part of the packet forwarding set by considering a circular region of radius R/2 centered at mid point of the particular pair of black nodes. If node is inside this region, it becomes an active forwarding node for that cluster pair with some random delay.
- When it becomes a forwarding node it sends a packet to signal this event. All other nodes go into sleep mode for that pair of clusters.
- A node may give up its active state for a cluster pair when it has spent a certain amount of energy. It sends a signal so that one of the other sleeping nodes can take over. When it gets a signal back from some node it goes to sleep mode for that cluster pair.
- The intermediate node between two black nodes is used for forwarding if overlap region does not have nodes in both clusters.
- While forwarding a black node listens to all packets and forwards only if the sending node from is out of range of the active forwarding node.

5. SIMULATIONS AND RESULTS

In this section we analyze the performance of our proposed schemes. We modified the NS-2 simulator to incorporate details of topology discovery algorithms. We assume that sensors are randomly deployed in a field of 100m x 100m. The number of sensors and the communication range of the sensors vary according to the experiment requirements. The results represent an average of these values for ten different runs.

Our first experiment is designed to find the number of black nodes formed during a topology discovery, when the number of nodes increases. We compare our results against the centralized log(n)-approximate solution provided by the greedy algorithm for set cover [9]. Figure 3 and 4 shows the number of black nodes formed at the end of *TopDisc* and the corresponding values for the centralized algorithm for varying number of nodes and communication range respectively. We observe that our algorithm works nearly as well as the centralized algorithm.

Next we evaluate the overhead incurred in doing the topology discovery. We compare the TopDisc approach against direct response and aggregated response approaches. Overhead is characterized by the number of packets and the number of bytes transmitted during the entire operation. Since the request propagation phase is same for all the approaches, the graphs show the overhead incurred during the response phase only. The results are evaluated with varying number of nodes (with communication range 20m) and varying communication range (with number of nodes 200). In evaluating the number of bytes, we assume a constant packet header of 5 bytes and an additional 5 bytes of information per node reported in the packet. Figure 5 shows the byte overhead of TopDisc compared to the trivial approaches for increasing number of nodes. Since the number of black nodes is almost constant with respect to number of nodes, the overhead of TopDisc increases marginally and hence has significantly lesser overhead than the trivial approaches.

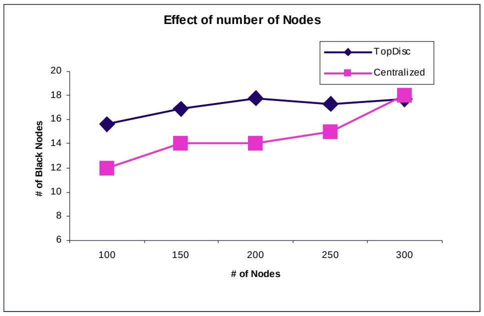

Figure 3. Number of Black nodes vs total number of nodes. Range 20m

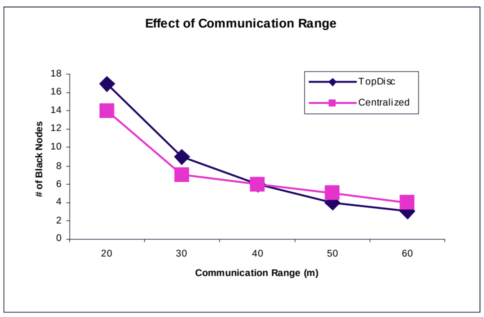

Figure 4. Number of Black nodes vs communication range. 200 nodes

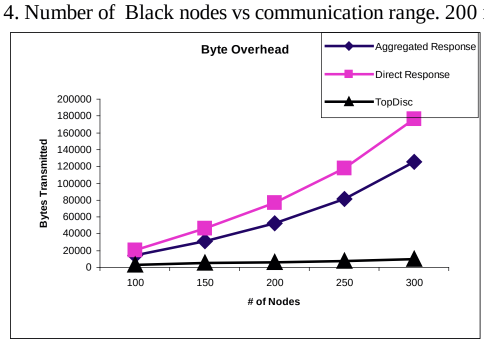

Figure 5. Byte Overhead for topology discovery, range 20m

The routing approach based on forwarding between clusters provides us a way of distributing load among different nodes. Figure 6 gives the number of nodes, which are involved in sharing the load in each cluster. Overall it shows that in all cases nearly 50% of the total nodes are part of the forwarding set although the minimum number required for reachability is much lower (number of black nodes + number of default nodes).

6. CONCLUSIONS AND FUTURE WORK

In this paper we have described a topology discovery algorithm *(TopDisc)* for wireless sensor networks. *TopDisc* selects a set of distinguished nodes, and constructs a reachability map based on their information. *TopDisc* logically organizes the network in the form of clusters and forms a Tree of Clusters *(TreC)* rooted at the monitoring node. We showed the applications of *TreC* for efficient data dissemination and aggregation, duty cycle assignments and network state retrieval. *TopDisc* is completely distributed, uses only local information and is highly scalable.

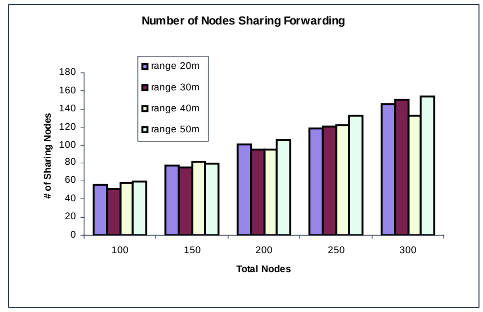

Figure 6. # of nodes sharing forwarding duty using TopDisc's duty cycle setting mechanism

This work presents a preliminary investigation into various aspects of topology discovery for sensor networks. In future, we would further investigate the various performance parameters that can be deduced using retrieved information. The final goal of this work is to define a sensor network management framework similar to SNMP [7] used for conventional networks.

7. REFERENCES

[1] J.M. Kahn, R.H. Katz, K.S.J. Pister, *Next century challenges: Mobile networking for 'smart dust*', Proc. MOBICOM, 1999, Seattle, 271-278

[2] Jaap Haartsen, Mahamoud Naghshineh, Joh Inouye, Oalf J. Joeressen, and Warren Allen, "*Bluetooth: Vision, Goals, and Architecture*," Mobile Computing and Communications Review, Oct. 1998, 2(4):38-45

[3] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. *System architecture directions for networked sensors*. In Proceedings of the 9th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, pages 93-104, Cambridge, Massachusetts, Nov. 2000

[4] W. Heinzelman, J. Kulik, and H. Balakrishnan; *Adaptive protocols for information dissemination in wireless sensor networks;* in Proceedings of Mobicom '99, pp. 174-185, 1999.

[5] Wendi Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, *Energy-Efficient Communication Protocols for Wireless Microsensor Networks*, Proc. Hawaaian Int'l Conf. on Systems Science, January 2000.

[6] C. Intanagonwiwat, R. Govindan and D. Estrin; *Directed diffusion:A scalable and robust communication paradigm for sensor networks*; in Proceedings of Mobicom '00, 2000.

[7] J. D. Case, M. Fedor, M. L. Schostall, and C. Davin. *RFC 1157: Simple network management protocol (SNMP*). RFC, IETF, May 1990

[8] A. Downey, "*Using pathchar to estimate Internet link characteristics*, Proc. SIGCOMM 1999, Cambridge, MA, pp. 241-250, Sept. 1999

[9] Cormen, Leiserson, Rivest, *Introduction to Algorithms*, MIT Press, McGraw Hill, 1990.